Enhancing Facial Features Detection with Deformable Face Templates

Levente Sajó

In research of multi-modal human-computer interaction, an important problem is to extract information from human faces. For this, in every case, the first step is to localize faces, then the localized face can be used in further processing steps, like gender, age, gesture recognition, facial feature extraction.

Because color images do not carry much more information, detection steps are made on grayscale images. Faces and facial features are complex shapes having high variability in scale and texture therefore their localization is not a simple task. Many different object detection techniques have been published in the last few years, the most successful detectors are template based and appearance based detectors [1]. Viola and Jones have developed a template based object detector (called Boosted Cascade Detector - BCD) which has a similar detection rate then the other detectors (based on SVM, Neural Networks) but needs much less computational time [2].

Our aim was to detect faces and facial features in real time, on video streams, so BCD was a plausible choice. BCD was used for localizing faces and on the localized face individual feature detectors were used to localize facial features (e.g. mouth and eyes corners). Face Detection showed us a convincing performance, but since facial features contains less information then the whole face, individual feature detectors seemed to be unreliable, therefore some further processing needed.

For enhancing the reliability of feature detectors two additional steps have been implemented. First, the search area of every individual feature detector has been restricted. In the second step, the detected feature points are checked whether fulfill a predefined shape constraint. For this, a face template has been defined, which contains template points corresponding to the facial features (we used 16 points, 4x3 point from eyebrows and eyes and 4 points from mouth). The relation between template-points is described with a one or more template rules. A rule defines the relative position of two or more template points. E.g.: one rule can define the symmetry of the face; the center of mouth points should lay on the perpendicular bisector of two eyes.

Every rule has a kernel function and three parameters: estimated value, tolerance, increment. We currently use two types of kernel functions, every type having two versions: one operates on x coordinate, the other on y coordinate of a point. One kernel function calculates the distance between two points (1); the other calculates the distance between mean of two points and a third point (2).

$$\begin{aligned} xDistance(P1, P2) &= P2.x - P1.x, \\ yDistance(P1, P2) &= P2.y - P1.y \end{aligned} \tag{1}$$

$$\begin{aligned} xMiddle(P1, P2, P3) &= (P1.x + P2.x)/2 - P3.x, \\ yMiddle(P1, P2, P3) &= (P1.y + P2.y)/2 - P3.y \end{aligned} \tag{2}$$

The penalty value is calculated as a combination of the return value of the kernel function and the other three rule parameters. If the kernel-function value is in the neighbor of the estimated value, the return value is 0, otherwise penalty value is calculated using increment value.

if (funcvalue is between [estimated value - tolerance, estimatedValue + tolerance])
then penaltyValue = 0;
else penaltyValue = abs(funcvalue - estimatedValue) - tolerance) * increment

The sum of penalty values gives the overall penalty value of a set of feature point. Every combination of the first few feature points found by individual detectors are matched to the face template and that set of feature points is selected which has the smallest penalty value. On the selected set of points, using the template, those points which has still large penalty value, it is possible to calculate a new (better) position, which together with other points forms a better face template.

In this stage of the project, the parameters of the rules are defined manually. Later a learning algorithm can be implemented, which uses a face database with manually defined face points and automatically defines the rules and their parameter.

References

[1] M. H. Yang, D. J. Kriegman, N. Ahuja. Detecting Faces in Images: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 24, January 2002.

[2] P. Viola, M. Jones. Robust Real-time Object Detection, Technical Report, *Cambridge Research Laboratory*, 2001.

[3] D. Cristinacce. Automatic Detection of Facial Features in Grey Scale Images, PhD Thesis, *University of Manchester*, 2004.